1. Amplitude Modulation (AM) Mathematics
2. Pitch Shifter with Single-Sideband AM
3. [mini-project] Ring Modulation and Pitch Shifting
4. Frequency Modulation (FM) Mathematics
5. Frequency Modulation (FM) Techniques in LabVIEW
6. Chowning FM Synthesis Instruments in LabVIEW
7. [mini-project] Chowning FM Synthesis Instruments

Amplitude Modulation (AM) Mathematics
Amplitude modulation (AM) creates interesting special effects when applied to music and speech signals. The mathematics of the modulation property of the Fourier transform are presented as the basis for understanding the AM effect, and several audio demonstrations illustrate the AM effect when applied to simple signals (sinusoids) and speech signals. The audio demonstration is implemented by a LabVIEW VI using an event structure as the basis for real-time interactive parameter control.

	This module refers to LabVIEW, a software development environment that features a graphical programming language. Please see the LabVIEW QuickStart Guide module for tutorials and documentation that will help you:
	• Apply LabVIEW to Audio Signal Processing
	• Get started with LabVIEW
	• Obtain a fully-functional evaluation edition of LabVIEW

Overview

Amplitude modulation (**AM**) is normally associated with communications systems; for example, you can find all sorts of "talk radio" stations on the AM band. In communication systems, the **baseband** signal has a bandwidth similar to that of speech or music (anywhere from 8 kHz to 20 kHz), and the modulating frequency is several orders of magnitude higher; the AM radio band is 540 kHz to 1600 kHz.

When applied to audio signals for music synthesis purposes, the modulating frequency is of the same order as the audio signals to be modulated. As described below, AM (also known as **ring modulation**) splits a given signal

spectrum in two, and shifts one version to a higher frequency and the other version to a lower frequency. The modulated signal is the sum of the frequency-shifted spectra, and can provide interesting special effects when applied to speech and music signals.

Modulation Property of the Fourier Transform

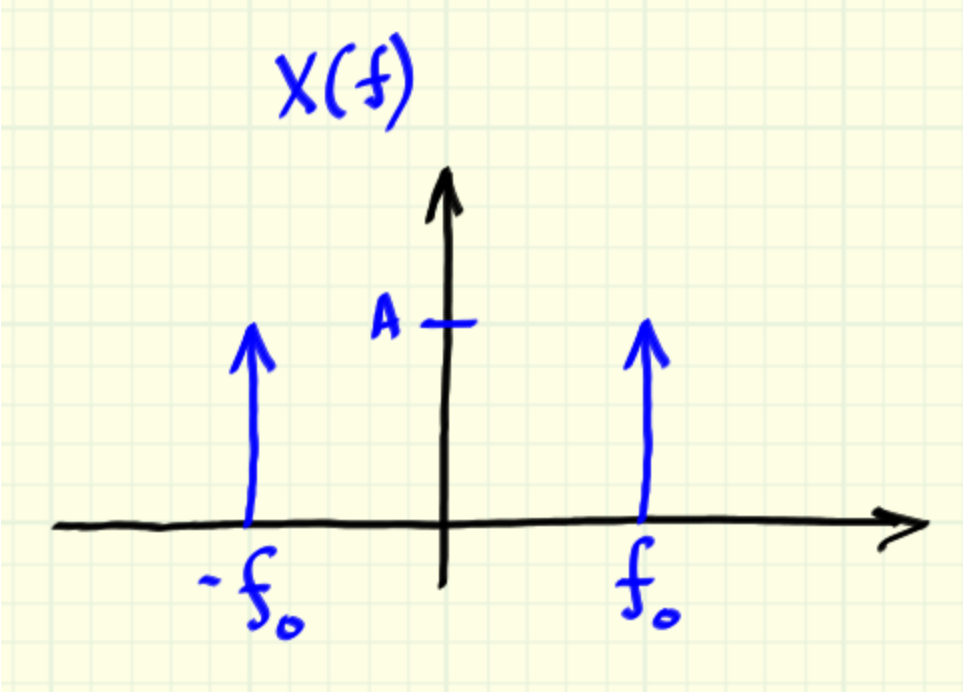
The **modulation property** of the Fourier transform forms the basis of understanding how AM modifies the spectrum of a source signal. The screencast video of [link] explains the modulation property concept and derives the equation for the modulation property.
https://youtu.be/ouajJJOn9Hg (4:38)
https://www.youtube.com/embed/ouajJJOn9Hg?rel=0

[video] Modulation property concepts and derivation

Suppose the source signal to be modulated contains only one spectral component, i.e., the source is a sinusoid. The screencast video of [link] shows how to apply the modulation property to predict the spectrum of the modulated signal. Once you have studied the video, try the exercises below to ensure that you understand how to apply the property for a variety of different modulating frequencies.
https://youtu.be/afHa5RZJ_X8 (2:31)
https://www.youtube.com/embed/afHa5RZJ_X8?rel=0

[video] Determine the spectrum of a modulated sinusoid

The time-domain signal is a sinusoid of amplitude with corresponding frequency-domain spectrum as shown in [link].



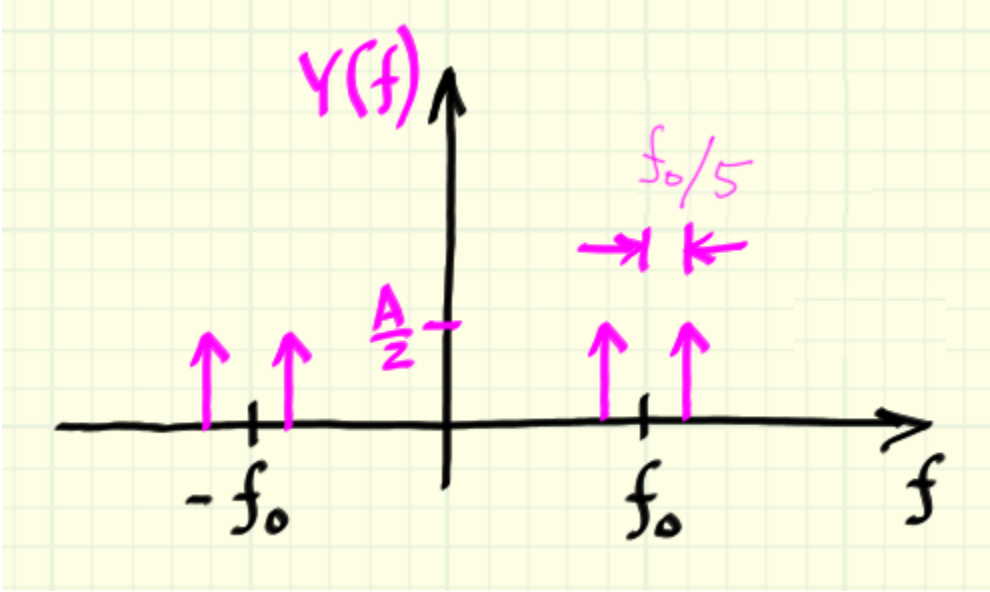
Spectrum of the signal x(t)

Suppose is modulated by a sinusoid of frequency . For each of the exercises below, draw the spectrum of the modulated signal , where the exercise problem statement indicates the modulation frequency.

Exercise:

Problem: fm = f0/5

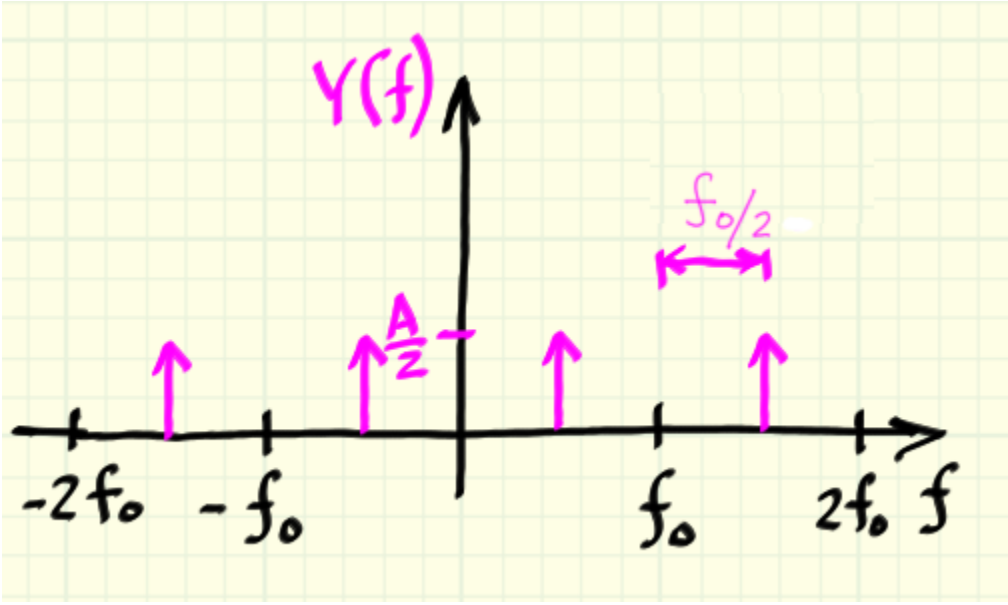
Solution:



Exercise:

Problem: fm = f0/2

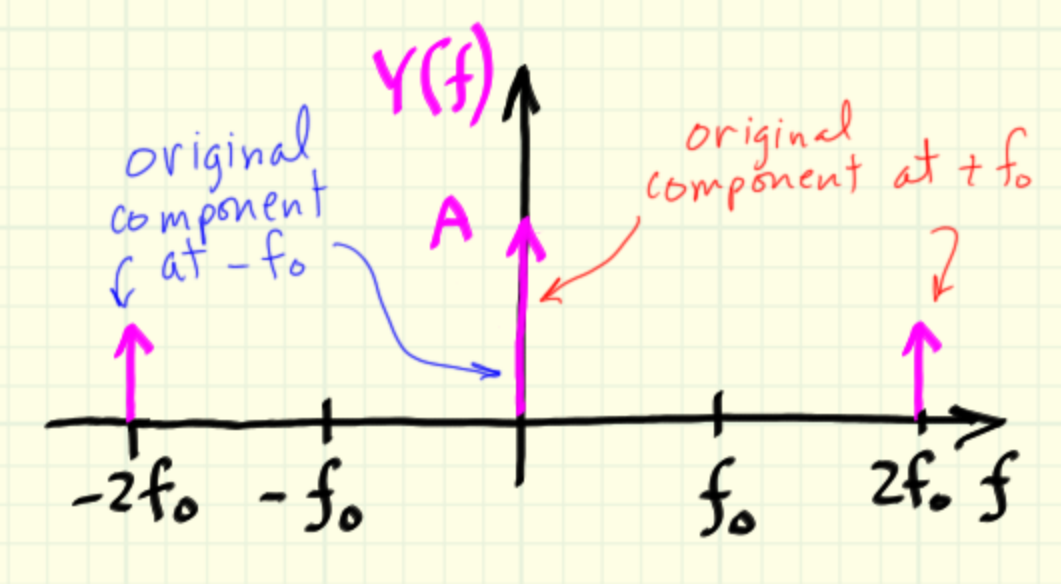
Solution:



Exercise:

Problem: fm = f0

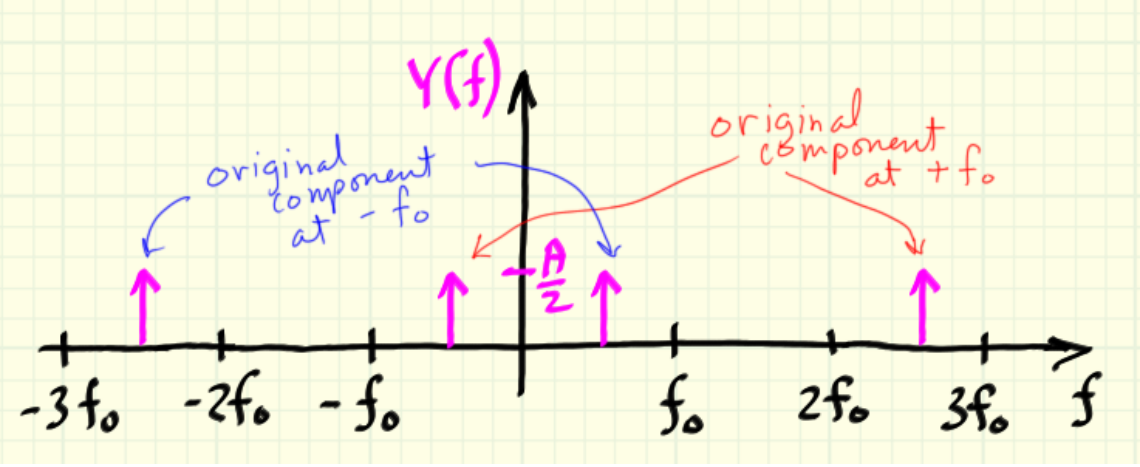
Solution:



Exercise:

Problem: fm = 1.5f0

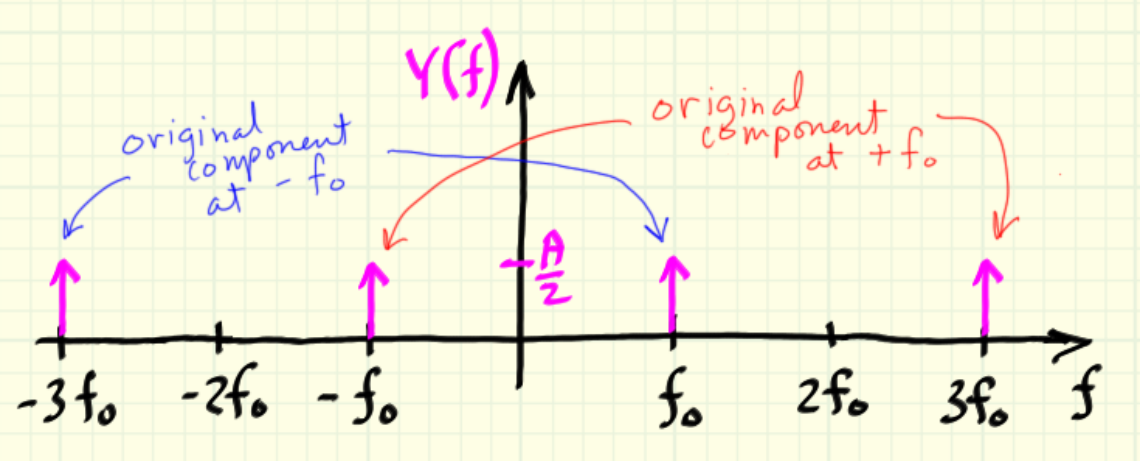
Solution:



Exercise:

Problem: fm = 2f0

Solution:



Did you notice something interesting when becomes larger than ? The right-most negative frequency component shifts into the positive half of the spectrum, and the left-most positive frequency component shifts into the negative half of the spectrum. This effect is similar to the idea of **aliasing**, in which a sinusoid whose frequency exceeds half the sampling frequency is said to be "folded back" into the principal alias. In the case of AM, modulating a sinusoid by a frequency greater than its own frequency folds the left-most component back into positive frequency.

Audio Demonstrations

The screencast video of [link] demonstrates the aural effects of modulating a single spectral component, i.e., a sinusoid. The LabVIEW code for the demo is also described in detail, especially the use of an **event structure** contained in a **while-loop structure** (see video in [link]). The event structure provides an efficient way to run an algorithm with real-time interactive parameter control without polling the front panel controls. The

event structure provides an alternative to the polled method described in Real-Time Audio Output in LabVIEW.

The LabVIEW VI demonstrated within the video is available here: am_demo1.vi. Refer to TripleDisplay to install the front-panel indicator used to view the signal spectrum.
https://youtu.be/vQ7sh1u5oYo (4:24)
https://www.youtube.com/embed/vQ7sh1u5oYo?rel=0

[video] Modulating a single sinusoid

https://youtu.be/G1UbdOsldR4 (6:39)
https://www.youtube.com/embed/G1UbdOsldR4?rel=0

[video] LabVIEW implementation of AM demo
using **event structure**

The next screencast video (see [link]) demonstrates the aural effects of modulating two spectral components created by summing together a sinusoid at frequency f0 and another sinusoid at frequency 2f0. You can obtain interesting effects depending on whether the spectral components end up in a harmonic relationship; if so, the components fuse together and you perceive a single pitch. If not, you perceive two distinct pitches.

The LabVIEW VI demonstrated within the video is available here: am_demo2.vi. Refer to TripleDisplay to install the front-panel indicator used to view the signal spectrum.
https://youtu.be/wuzEHT8zhBw (4:18)
https://www.youtube.com/embed/wuzEHT8zhBw?rel=0

[video] Modulating a pair of sinusoids

The third demonstration (see [link]) illustrates the effect of modulating a music clip and a speech signal. You can obtain Interesting special effects because the original source spectrum simultaneously shifts to a higher and lower frequency.

The LabVIEW VI demonstrated within the video is available here: am_demo3.vi. Refer to TripleDisplay to install the front-panel indicator used to view the signal spectrum.

The two audio clips used in the example are available here: flute.wav and speech.wav (speech clip courtesy of the Open Speech Repository, www.voiptroubleshooter.com/open_speech; the sentences are two of the many phonetically-balanced **Harvard Sentences**, an important standard for the speech processing community).
https://youtu.be/ytEYQz0XR5A (7:41)
https://www.youtube.com/embed/ytEYQz0XR5A?rel=0

[video] Modulating a music clip and a speech signal

Pitch Shifter with Single-Sideband AM
Pitch shifting makes an interesting special effect, especially when applied to a speech signal. Single-sideband amplitude modulation (SSB-AM) is presented as a method to shift the spectrum of a source signal in the same way as basic AM, but with cancellation of one sideband to eliminate the “dual voice” sound of conventional AM. Pre-filtering of the source signal to avoid aliasing is also discussed.

	This module refers to LabVIEW, a software development environment that features a graphical programming language. Please see the LabVIEW QuickStart Guide module for tutorials and documentation that will help you:
	• Apply LabVIEW to Audio Signal Processing
	• Get started with LabVIEW
	• Obtain a fully-functional evaluation edition of LabVIEW

Overview

Amplitude modulation (**AM**) of a source signal divides the signal's spectrum into two copies, with one copy shifted towards higher frequency and the other copy shifted towards lower frequency; refer to AM Mathematics for a complete treatment of basic AM. The shifted and dual spectrum makes an interesting special effect when applied to a musical instrument or the human voice, creating the sensation of two different people speaking the identical phrase, for example.

If one of these spectral images could somehow be cancelled out, AM seems to be a feasible way to implement a **pitch shifter**, a device or algorithm that shifts the source spectrum higher or lower in frequency. When this special

effect is applied in real time, you can speak into a microphone and sound just like one of "Alvin and the Chipmunks."

As an example of what you will be able accomplish by applying the techniques presented in this module, listen to this original speech clip speech.wav and its pitch-shifted version speech_shifted.wav (speech clip courtesy of the Open Speech Repository, www.voiptroubleshooter.com/open_speech; the sentences are two of the many phonetically balanced **Harvard Sentences**, an important standard for the speech processing community).

Single-Sideband AM (SSB-AM)

The screencast video of [link] develops the basic theory of **single-sideband (SSB) modulation**, a technique borrowed from communications systems that provides a way to apply amplitude modulation with spectral image cancellation.
https://youtu.be/aqS7q6RfOKM (7:48)
https://www.youtube.com/embed/aqS7q6RfOKM?rel=0

[video] Single-sideband modulation for pitch shifting

As an exercise to ensure that you followed each step, draw a block diagram or flow diagram to show how the original signal is modified to produce the final shifted signal. Your diagram should include directed lines (arrows) to show signal flow, and should use symbols (blocks) for the multipliers, cosine and sine oscillators, Hilbert transformer, sign changer, and adder.

Pre-Filtering to Avoid Aliasing

Pre-filtering the source signal ensures the shifted spectrum does not alias, since the source signal typically fills the available bandwidth. The screencast video of [link] discusses the aliasing problem as well as the techniques you can use to design a suitable pre-filter.

https://youtu.be/M8NfbbJABWo (3:18)
https://www.youtube.com/embed/M8NfbbJABWo?rel=0

[video] Pre-filtering to avoid aliasing

[mini-project] Ring Modulation and Pitch Shifting

	This module refers to LabVIEW, a software development environment that features a graphical programming language. Please see the LabVIEW QuickStart Guide module for tutorials and documentation that will help you:
	• Apply LabVIEW to Audio Signal Processing
	• Get started with LabVIEW
	• Obtain a fully-functional evaluation edition of LabVIEW

Overview

Ring modulation (AM) is an audio special effect that produces two frequency-shifted replicas of the spectrum of a source signal, with one replica shifted to higher frequency and the other replica to a lower frequency. **Single-sideband AM (SSB-AM)** provides a way to shift the source signal's spectrum higher or lower but without the additional replica. SSB-AM provides one way to implement a **pitch shifter**, an audio special effect that shifts the frequency of speech, singing, or a musical instrument to a higher or lower frequency.

In this project, use LabVIEW to implement several types of ring modulators and a pitch shifter.

Prerequisite Modules

If you have not done so already, please study the prerequisite modules AM Mathematics and Pitch Shifting. If you are relatively new to LabVIEW,

consider taking the course LabVIEW Techniques for Audio Signal Processing which provides the foundation you need to complete this mini-project activity, including working with arrays, creating subVIs, playing an array to the soundcard, and saving an array as a .wav sound file.

Deliverables

- All LabVIEW code that you develop (block diagrams and front panels)
- All generated sounds in .wav format
- Any plots or diagrams requested
- Summary write-up of your results

Part 1: Multiple Modulators

Consider an original signal , which is a sinusoid of frequency . The original signal is modulated by a cosine function of frequency to produce , which is in turn modulated by a cosine function of frequency to produce , which is in turn modulated by a cosine function of frequency to produce . Sketch the frequency-domain version of the four signals, i.e., sketch , , , and .

Create a LabVIEW implementation of the above arrangement and plot the spectrum of each of the four signals. Compare your LabVIEW results to your prediction.

Part 2: Multiple Modulators with Soundfile Input

Create a LabVIEW implementation of the multiple modulation scheme of Part 1 that can process a .wav audio file as the input signal. Use controls for the three modulators that will allow you to easily change their modulation frequencies. Experiment with various choices of modulation frequencies to make an interesting effect. Create two .wav files using different parameter choices.

Part 3: Pitch Shifter

Implement the pitch shifting algorithm based on the single-sideband AM technique discussed in Pitch Shifter with Single-Sideband AM. Use a design similar to that of "am_demo3.vi" provided at the bottom of the page of AM Mathematics which accepts a .wav file as input and plays the sound. The sound clip should be relatively short (on the order of several seconds). For this part of the project, do **not** implement the pre-filter; you will do this in Part 4.

Evaluate the quality of your pitch shifter by presenting some written discussion and suitable spectrogram plots. Especially indicate whether you can find audible and visual evidence of aliasing.

The **fast Hilbert transform** built-in subVI is available in the "Signal Processing | Transforms" pallet.

Part 4: Pitch Shifter with Anti-Aliasing Filter

Modify your pitch shifter to include a bandpass filter. State how you will compute the bandpass filter's upper and lower corner frequencies, given that you want to preserve as much of the original signal's bandwidth as possible.

Evaluate the quality of your modified pitch shifter by presenting some written discussion and suitable spectral plots. Compare your results with those you obtained in Part 3.

A variety of digital filters are available in the "Signal Processing | Filters" pallet.

Optional Part 5: Real-Time Processor

Choose one of the previous LabVIEW implementations and make it work in real time with a signal input (microphone) and interactive front-panel controls.

Evaluate the interrupt-driven approach using an event structure (see "am_demo1.vi" described in AM Mathematics, as well as the polled approach used by mic_in_speaker_out.vi). Use whichever technique you prefer.

Submit your finished LabVIEW implementation as a distinct .zip file.

Frequency Modulation (FM) Mathematics

Frequency modulation (FM) in the audio frequency range can create very rich spectra from only two sinusoidal oscillators, and the spectra can easily be made to evolve with time. The mathematics of FM synthesis is developed, and the spectral characteristics of the FM equation are discussed. Audio demonstrations as implemented by LabVIEW VIs illustrate the relationships between the three fundamental FM synthesis parameters (carrier frequency, modulation frequency, modulation index) and the synthesized spectra.

	This module refers to LabVIEW, a software development environment that features a graphical programming language. Please see the LabVIEW QuickStart Guide module for tutorials and documentation that will help you:
	• Apply LabVIEW to Audio Signal Processing
	• Get started with LabVIEW
	• Obtain a fully-functional evaluation edition of LabVIEW

Overview

Frequency modulation (**FM**) is most often associated with communications systems; for example, you can find all sorts of music stations on the FM band of your radio. In communications systems the **baseband** signal has a bandwidth similar to that of speech or music (anywhere from 8 kHz to 20 kHz), and the modulating frequency is several orders of magnitude higher; the FM radio band is 88 MHz to 108 MHz.

When applied to audio signals for music synthesis purposes, the modulating frequency is of the same order as the audio signals to be modulated. FM can create very rich spectra, and the spectra can easily be made to evolve with time. The ability of FM to produce a wide variety of interesting spectra from only two sinusoidal oscillators makes FM a fascinating synthesis technique.

Brief History of FM Synthesis

John Chowning was the first to systematically evaluate FM in the audio spectrum, and along with Stanford University, filed for a patent on the technique in 1975 (see U.S. Patent 4,018,121 at U.S. Patent and Trademark Office or at Google Patent Search). The patent was issued in 1977, and Stanford University licensed the technology to Yamaha Corporation. Six years later in 1983, Yamaha introduced the revolutionary DX7 synthesizer ([link]), the first commercially successful instrument based on FM synthesis. The DX7 was also a milestone by introducing two other new technologies: digital synthesis and MIDI (Musical Instrument Digital Interface). The "FM sound" defines much of the pop music styles of the 1980s.

Yamaha DX7 synthesizer, the first commercially successful instrument to offer FM synthesis, digital synthesis, and MIDI compatibility. The instrument pictured here is packaged in a road case. Photographer:

schoschie (http://www.flickr.com/photos/schoschie/51653026/). Copyright holder has granted permission to display this image under the Creative Commons Attribution-ShareAlike license.

FM Equation

The basic FM equation is presented in [link]:
Equation:

where the parameters are defined as follows:

- carrier frequency (Hz)
- modulation frequency (Hz)
- modulation index

The [link] screencast video continues the discussion by explaining the significance of each part of [link], and demonstrates in a qualitative fashion how the different parameters of the equation influence the spectrum of the audio signal.

Download the LabVIEW VI demonstrated in the video: fm_demo1.vi. Refer to TripleDisplay to install the front-panel indicator used to view the signal spectrum.
https://youtu.be/49SOGF4AUIY (3:16)
https://www.youtube.com/embed/49SOGF4AUIY?rel=0

[video] Significance of each part of the basic FM equation, and audio demonstration

FM Spectrum

The following trigonometric identity facilitates quantitative understanding of the spectrum produced by the basic FM equation of [link]:

Equation:

The term defines a Bessel function of the first kind of order evaluated at the value .

Note how the left-hand side of the identity possesses exactly the same form as the basic FM equation of [link]. Therefore, the right-hand side of the identity explains where the spectral components called **sidebands** are located, and indicates the amplitude of each spectral component. The [link] screencast video continues the discussion by explaining the significance of each part of [link], especially the location of the sideband spectral components.
https://youtu.be/33skRAysD0g (4:59)
https://www.youtube.com/embed/33skRAysD0g?rel=0

[video] Trig identity of [link] and location of sideband spectral components

As discussed [link] video, the basic FM equation produces an infinite number of sideband components; this is also evident by noting that the summation of [link] runs from k=1 to infinity. However, the amplitude of each sideband is controlled by the Bessel function, and non-zero amplitudes tend to cluster around the central carrier frequency. The [link] screencast video continues the discussion by examining the behavior of the Bessel function when its two parameters are varied, and shows how these parameters link to the modulation index and sideband number.
https://youtu.be/RrqMiS-yMCI (8:21)
https://www.youtube.com/embed/RrqMiS-yMCI?rel=0

[video] Discussion of the Bessel function and its relationship to modulation index and sideband number

Now that you have developed a better quantitative understanding of the spectrum produced by the basic FM equation, the [link] screencast video revisits the earlier audio demonstration of the FM equation to relate the spectrum to its quantitative explanation.
https://youtu.be/XCtVR1_xvSw (3:51)
https://www.youtube.com/embed/XCtVR1_xvSw?rel=0

[video] FM audio demonstration revisited

Harmonicity Ratio

The basic FM equation generates a cluster of spectral components centered about the carrier frequency

with cluster density controlled by the modulation frequency

. Recall that we perceive multiple spectral components to be a single tone when the components are located at integer multiples of a fundamental frequency, otherwise we perceive multiple tones with different pitches. The **harmonicity ratio** provides a convenient way to choose the modulation frequency to produce either harmonic or inharmonic tones. Harmonicity ratio is defined as:
Equation:

Harmonicity ratios that involve an integer, i.e., or for , result in sideband spacing that follows a harmonic relationship. On the other hand, non-integer-based harmonicity ratios, especially using irrational numbers such as and , produce interesting inharmonic sounds.

Try experimenting with the basic FM equation yourself. The LabVIEW VI fm_demo2.vi provides front-panel controls for carrier frequency, modulation index, and harmonicity ratio. You can create an amazingly wide variety of sound effects by strategically choosing specific values for these three parameters. The [link] screencast video illustrates how to use the VI and provides some ideas about how to choose the parameters. Refer to TripleDisplay to install the front-panel indicator used to view the signal spectrum.
https://youtu.be/U1RIpCMRW_0 (1:54)
https://www.youtube.com/embed/U1RIpCMRW_0?rel=0

[video] Demonstration of fm_demo2.vi

References

- Moore, F.R., "Elements of Computer Music," Prentice-Hall, 1990, ISBN 0-13-252552-6.
- Dodge, C., and T.A. Jerse, "Computer Music: Synthesis, Composition, and Performance," 2nd ed., Schirmer Books, 1997, ISBN 0-02-864682-7.

Frequency Modulation (FM) Techniques in LabVIEW
Frequency modulation synthesis (FM synthesis) creates a rich spectrum using only two sinusoidal oscillators. Implementing the basic FM synthesis equation in LabVIEW requires a special technique to make one oscillator vary the phase function of the other oscillator. Learn how to implement the basic FM equation, and also hear an audio demonstration of the equation in action.

	This module refers to LabVIEW, a software development environment that features a graphical programming language. Please see the LabVIEW QuickStart Guide module for tutorials and documentation that will help you:
	• Apply LabVIEW to Audio Signal Processing
	• Get started with LabVIEW
	• Obtain a fully-functional evaluation edition of LabVIEW

Overview

Frequency modulation synthesis (**FM synthesis**) creates a rich spectrum using only two sinusoidal oscillators; refer to FM Mathematics for a complete treatment of the mathematics of FM synthesis. Implementing the basic FM synthesis equation in LabVIEW requires a special technique to make one sinusoidal oscillator vary the **phase function** of the other sinusoidal oscillator. The following screencast video walks you through the implementation process of the basic FM equation and includes an audio demonstration of the equation in action.

Refer to TripleDisplay to install the front-panel indicator used to view the signal spectrum.

https://youtu.be/BS1PH5GUHh8 (10:23)
https://www.youtube.com/embed/BS1PH5GUHh8?rel=0

[video] Implementing the basic FM synthesis equation in LabVIEW

Chowning FM Synthesis Instruments in LabVIEW
John Chowning pioneered frequency modulation (FM) synthesis in the 1970s, and demonstrated how the technique could simulate a diversity of instruments such as brass, woodwinds, and percussion. FM synthesis produces rich spectra from only two sinusoidal oscillators, and more interesting sounds can be produced by using a time-varying modulation index to alter the effective bandwidth and sideband amplitudes over time. A LabVIEW VI is developed to implement the sound of a clarinet, and the VI can be easily modified to simulate the sounds of many other instruments.

	This module refers to LabVIEW, a software development environment that features a graphical programming language. Please see the LabVIEW QuickStart Guide module for tutorials and documentation that will help you:
	• Apply LabVIEW to Audio Signal Processing
	• Get started with LabVIEW
	• Obtain a fully-functional evaluation edition of LabVIEW

Overview

Frequency modulation synthesis (**FM synthesis**) produces incredibly rich spectra from only two sinusoidal oscillators; refer to FM Mathematics for a complete description of the spectral characteristics of FM synthesis. You can produce even more interesting sounds with a time-varying modulation index to alter the effective bandwidth and sideband amplitudes over time. This relatively simple modification to the basic FM equation creates tones with time-varying spectra to emulate many types of physical musical instruments.

John Chowning pioneered FM synthesis in the 1970s and demonstrated how the technique could simulate instruments such as brass, woodwinds, and percussion. These techniques are the subject of this module.

Significance of Time-Varying Spectra

Physical musical instruments produce audio spectra that evolve with time. A typical sound begins with some type of dynamic transient, for example, as pressure builds up within a brass instrument or when a percussion instrument is first struck. The sound continues with some type of quasi steady-state behavior when mechanical energy is continually applied, i.e., blowing on a flute, bowing a violin string, and repeatedly striking a gong. Once the mechanical energy input ceases, the sound concludes by decaying in some fashion to silence.

Clearly the amplitude of the instrument's audio signal changes during the course of the tone, following the typical attack-decay-sustain-release (ADSR) envelope described in Analog Synthesis Modules. Even more important, the **intensity** of the higher-frequency spectral components changes as well. The high-frequency components are often more evident during the initial transient. In fact, the dynamic nature of the spectra during the instrument's transient plays an important role in timbre perception.

FM Equation with Time-Varying Modulation Index

The basic FM equation with time-varying amplitude and modulation index is presented in [link]:
Equation:

You can easily model a physical instrument with this equation by causing the modulation index to **track** the time-varying amplitude . In this way, a louder portion of the note also has more sidebands, because the modulation index effectively controls the bandwidth of the FM spectra.

Chowning Instruments

John Chowning's publication, *The Synthesis of Complex Audio Spectra by Means of Frequency Modulation (Journal of the Audio Engineering Society, 21(7), 1973)*, describes a basic structure to implement [link] with the following parameters:

- - peak amplitude
- - maximum value of modulation index i(t)
- - minimum value of modulation index
- [Hz] - carrier frequency
- - harmonicity ratio ( )
- duration [s] - duration of generated audio
- - prototype waveform for time-varying amplitude
- - prototype waveform for time-varying modulation index

The prototype waveforms are normalized in both dimensions, i.e., the range and domain are both zero to one. The prototype waveform is converted to the time-varying amplitude as . The prototype waveform is converted to the time-varying modulation index as . Representative Chowning FM instrument specifications are described in the PDF document chowning_instruments.pdf. The [link] screencast video walks through the complete process to implement the Chowning clarinet instrument in LabVIEW.

Download the finished VI: chowning_clarinet.vi. Refer to TripleDisplay to install the front-panel indicator used to view the signal spectrum.

You can easily adapt the VI to create the remaining Chowning instruments once you understand the general implementation procedure.
https://youtu.be/tFJ1QT80eIw (17:22)
https://www.youtube.com/embed/tFJ1QT80eIw?rel=0

[video] Implement a Chowning FM instrument

in LabVIEW

[mini-project] Chowning FM Synthesis Instruments

	This module refers to LabVIEW, a software development environment that features a graphical programming language. Please see the LabVIEW QuickStart Guide module for tutorials and documentation that will help you:
	• Apply LabVIEW to Audio Signal Processing
	• Get started with LabVIEW
	• Obtain a fully-functional evaluation edition of LabVIEW

Objective

FM synthesis creates rich spectra from only two sinusoidal oscillators, and is able to emulate the sound of many physical musical instruments. John Chowning's seminal publication on audio-range frequency modulation (FM) in 1973 describes a number of different orchestral instruments such as woodwinds, brass, and percussion that can be created merely by adjusting a few basic parameters of the basic FM equation.

In this mini-project, implement several different Chowning FM instruments and compare them to the sounds of physical instruments. Also develop code to model the Chowning algorithms as LabVIEW **virtual musical instruments** (**VMIs**) to be "played" by a MIDI file within **MIDI JamSession**.

Prerequisite Modules

If you have not done so already, please study the prerequisite modules FM Mathematics and Chowning FM Synthesis Instruments in LabVIEW. If you are relatively new to LabVIEW, consider taking the course LabVIEW Techniques for Audio Signal Processing which provides the foundation you need to complete this mini-project activity, including working with arrays, creating subVIs, playing an array to the soundcard, and saving an array as a .wav sound file.

Deliverables

- All LabVIEW code that you develop (block diagrams and front panels)
- All generated sounds in .wav format
- Any plots or diagrams requested
- Summary write-up of your results

Part 1: Chowning FM Instruments

Chowning FM Synthesis Instruments in LabVIEW provides the specifications for a generic FM synthesis instrument, parameters for a number of different instruments, and a screencast video that walks through the complete process to implement the Chowning clarinet in LabVIEW. Refer to the PDF document in that module that contains the parameters for the remaining instruments: bell, wood-drum, brass, and bassoon. Create your own LabVIEW implementation of each of these four instruments (the clarinet VI is available in that module, as well).

Save a representative sound from each of the five Chowning instruments to a .wav file.

> **Note:** Consider using an audio editor such as Audacity to merge the individual .wav files into a single .wav file that you submit as your deliverable. You can also add your own voice annotation to explain your work.

Part 2: Comparison with Physical Instruments

Visit the **Musical Instrument Samples** database created by the Electronic Music Studios of the University of Iowa at http://theremin.music.uiowa.edu/MIS.html. These recordings of actual instruments were made inside an anechoic chamber to eliminate reflections and other artifacts, so the spectra of the instruments are as accurate as possible. The files are stored in AIFF format; use an audio editor such as Audacity to import the AIFF format. Audacity also includes a tool to view the spectra of the soundfile.

Compare and contrast the FM sounds you created for brass, clarinet, and bassoon to those of the real instruments. Consider time-domain envelope shape and spectrogram patterns.

Part 3: Chowning VMIs for MIDI JamSession

In this part, convert each of the five Chowning instruments you implemented in Part 1 into its own **virtual musical instrument** (**VMI** for short) that can be played by "MIDI JamSession." If necessary, visit MIDI JamSession, download the application VI .zip file, and view the screencast video in that module to learn more about the application and how to create your own VMI. Your VMI will accept parameters that specify frequency, amplitude, and duration of a single note, and will produce a corresponding array of audio samples.

You may wish to keep all of your existing front-panel controls available so that you can listen to your VMI during development. Adjust the parameters to obtain pleasing and realistic settings, and convert the front-panel controls to constants and remove all indicators. Your finished VMI must not contain any front-panel controls or indicators beyond those provided in the prototype instrument.

Finally, choose a suitable MIDI file and use MIDI JamSession to play your FM VMIs. MIDI files that contain multiple channels are ideal, because you can individually assign each of your five VMIs to a different instrument.

Create a .wav file of your finished work.

Note: MIDI percussion events are found on Channel 10, a reasonable place to use your wood-drum instrument. Be aware that the "frequency" value produced by the prototype VMI derives directly from the "note number" value of the MIDI "Note On" messages in the file. On Channel 10, the note number selects from a palette of different percussion instruments as defined in the **General MIDI Sound Set** (http://www.midi.org/about-midi/gm/gm1sound.shtml), so interpreting the value as frequency is meaningless. You can either set up your wood-drum to produce the same waveform independent of the frequency parameter, or you can devise a scheme to translate the note number into useful parameter change for your instruments.